***Loan Status Prediction:***

**Problem Statement**

* This is a Property Loan Status Prediction dataset, we have the data of applicants who had previously applied for the loan based on the property and applicant-based metrics.
* The bank will decide whether to give a loan to the applicant based on some factors such as Applicant Income, Loan Amount, previous Credit History, Co-applicant Income, etc.
* The objective is to build a Machine Learning Model to predict the loan to be Approved or to be rejected for an applicant.
* You are free to use any classification model such Logistic Regression, KNN or Decision Tree.

**Univariate analysis:**

* Univariate analysis is when we analyze each variable individually. For categorical features we can use frequency table or bar plots which will calculate the number of each category in a particular variable.

**Loan-Amount:-**

* The loan of 422 (around 69%) people out of 614 was approved. There is no imbalanced classes issue in this dataset, thus accuracy as an evaluation metric should be appropriate. On the other hand, if there are imbalanced or skewed classes, then we might need to use precision and recall as evaluation metrics**.**

**Categorical-Variable**

It can be inferred from the above bar plots that:

* 80% applicants in the dataset are male.
* Around 65% of the applicants in the dataset are married.
* Around 15% applicants in the dataset are self-employed.
* Around 85% applicants have credit history (repaid their debts).
* Around 80% of the applicants are Graduate.

**Independent-Variable**

* More than half of the applicants don’t have any dependents.
* Most of the applicants are from Semi-Urban area.

**Loan-Amount -** Around 85% of the loans are 360 months term or 30 years period

**Bivariate Analysis:-**

After looking at every variable individually in univariate analysis, we will now explore them again with respect to the target variable in bivariate analysis. We can use bivariate analysis to test the hypotheses that we generated earlier.

* proportion of male and female applicants is more or less same for both approved and unapproved loans
* proportion of married applicants is higher for the approved loans
* distribution of applicants with 1 or 3+ dependents is similar across both the categories of Loan Status
* There is nothing significant we can infer from Self-employed vs Loan Status plot.
* proportion of loans getting approved for graduates is higher compared to non-graduates
* it seems people with credit history as 1 are more likely to get their loans approved
* Proportion of loans getting approved in semi urban area is higher as compared to that in rural or urban areas.

**We see that the most correlated variables are**

* (Applicant Income - Loan Amount) with correlation coefficient of 0.57
* (Credit\_History - Loan Status) with correlation coefficient of 0.56
* Loan Amount is also correlated with CoapplicantIncome with correlation coefficient of 0.19.

**Data Pre-processing**

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data pre-processing is a method of resolving such issues.

**Missing value and outlier treatment:**

Now the distribution looks much closer to normal and effect of extreme values has been significantly subsided. Let’s build a logistic regression model and make predictions for the test dataset.

**Conclusion:**

After trying and testing 4 different algorithms, the best accuracy on the public leaderboard is achieved by

* Logistic Regression (0.7847)
* Random Forest (0.7778)
* XGBoost (0.7778),
* Decision Tree performed the worst (0.6458).

While new features created via feature engineering helped in predicting the target variable, it did not improve the overall model accuracy much. Compared to using default parameter values, GridSearchCV helped improved the model's mean validation accuracy by providing the optimized values for the model's hyper parameters. On the whole, a logistic regression classifier provides the best result in terms of accuracy for the given dataset, without any feature engineering needed. Because of its simplicity and the fact that it can be implemented relatively easy and quick, Logistic Regression is often a good baseline that data scientists can use to measure the performance of other more complex algorithms. In this case, however, a basic Logistic Regression has already outperformed other more complex algorithms like Random Forest and XGBoost, for the given dataset.

**Suggestions for Improvement.**

There are many things that can be tried to improve the models’ predictions. We can create and add more variables, try different models with different subset of features and/or rows, etc. Some of the ideas are listed below:

* Combine the applicants with 1, 2, 3 or more dependents and make a new feature as discussed in the EDA part.
* Make independent vs independent variable visualizations to discover some more patterns.
* Arrive at the EMI using a better formula which may include interest rates as well.
* Try ensemble modeling (combination of different models). More about ensemble techniques can be found at the references.
* Try neural network using Tensor flow or PyTorch